

Introduction

Maven

GA et la forge GitHub

Intégration continue

Merge requests

Références

Maven - Forge - Intégration continue

Matthieu Moy (Transparents originaux d'Emmanuel Coquery)

<https://matthieu-moy.fr/>

2020

Matthieu Moy (Transparents originaux d'Emmanuel Coquery)

Maven - Forge - Intégration continue

2020

< 1 / 46 >

Introduction

Maven

GA et la forge GitHub

Intégration continue

Merge requests

Références

Autour du développement

Au delà du code:

- Tests (unitaires, intégration, fonctionnels)
- Documentation
- Partage des sources
- Suivi de bugs / évolutions
- Qualité du code
- Distribution

→ cycles de développement lourds à gérer

Matthieu Moy (Transparents originaux d'Emmanuel Coquery)

Maven - Forge - Intégration continue

2020

< 2 / 46 >

Introduction

Maven

GA et la forge GitHub

Intégration continue

Merge requests

Références

Outils

- Frameworks de tests
- Générateurs de documentation
- Gestionnaires de version
- Gestionnaires de tickets
- Outils d'audit de code
- Scripts, *builders*

Matthieu Moy (Transparents originaux d'Emmanuel Coquery)

Maven - Forge - Intégration continue

2020

< 3 / 46 >

Introduction

Maven

GA et la forge GitHub

Intégration continue

Merge requests

Références

Maven

Automatisation du traitement des phases du cycle de vie

- Peut être vu comme un "super Makefile"
 - Java comme langage de script
- Lance l'exécution d'outils:
 - Compilation
 - Test automatisés
 - Archives, Déploiement
 - Génération de documentation
 - ...

Alternatives: CMake, Premake, Grunt, Gulp, etc

Matthieu Moy (Transparents originaux d'Emmanuel Coquery)

Maven - Forge - Intégration continue

2020

< 4 / 46 >

Introduction

Maven

GA et la forge GitHub

Intégration continue

Merge requests

Références

Architecture

- Basée sur un système de plugins
 - Un plugin ↔ un script Java
 - ★ i.e. une classe avec une méthode particulière
 - ★ paramétrable via un morceau de XML
 - Une exécution de maven ↔ suite d'exécution de plugins
- Nombreux plugins disponibles
 - Pas tous installés au départ
 - Système de téléchargement de plugins à la demande

Matthieu Moy (Transparents originaux d'Emmanuel Coquery)

Maven - Forge - Intégration continue

2020

< 5 / 46 >

Introduction

Maven

GA et la forge GitHub

Intégration continue

Merge requests

Références

Phases et cycles de vie

- Une phase regroupe un ensemble de tâches (goals)
 - 1 tâche → 1 plugin
- Un cycle de vie est une suite de phases
 - Le déclenchement d'une phase déclenche les phases précédentes du cycle de vie
- Le cycle de vie dépend du *packaging* (jar,war, ...)
 - *packaging* = type de projet
 - Format d'archive
 - Ordre des phases
 - Affectation tâches → phases
 - Préconfiguration des tâches
 - peut être reconfiguré selon les besoins du projet

Matthieu Moy (Transparents originaux d'Emmanuel Coquery)

Maven - Forge - Intégration continue

2020

< 6 / 46 >

Introduction

Maven

GA et la forge GitHub

Intégration continue

Merge requests

Références

Exemple: phases du *packaging* jar

Phase	Tâche(s)
process-resources	resources:resources
compile	compiler:compile
process-test-resources	resources:testResources
test-compile	compiler:testCompile
test	surefire:test
package	jar:jar
install	install:install
deploy	deploy:deploy

Matthieu Moy (Transparents originaux d'Emmanuel Coquery)

Maven - Forge - Intégration continue

2020

< 7 / 46 >

Introduction

Maven

GA et la forge GitHub

Intégration continue

Merge requests

Références

Projet maven: organisation des fichiers

- pom.xml ← config. du projet
- src/ ← sources
 - main/ ← à distribuer
 - ★ java/ ← code Java
 - ★ resources/ ← fichiers à distribuer (config appli, images, etc)
 - ★ webapp/ ← ressources web (pour les war: html, jsp, js, images)
 - ★ antlr4/ ← grammaire pour générer les *parsers*
 - ★ ...
 - test/ ← uniquement pour les tests
 - ★ java/, resources/, antlr4/, etc
- target/ ← tout ce qui est généré, il est supprimé par `clean`
il ne faut **pas** le versionner (utiliser `.gitignore`)

Matthieu Moy (Transparents originaux d'Emmanuel Coquery)

Maven - Forge - Intégration continue

2020

< 8 / 46 >

IntroductionMavenG8 et la forge G8LabIntégration continueMerge requestsRéférences

Projet Maven: le pom.xml

```
<project ...>
  ...
  <groupId>fr.univ-lyon1.info.ml</groupId>
  <artifactId>monProjet</artifactId>

  <dependencies>
    <dependency>...</dependency>
    <dependency>...</dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>...</plugin>
      <plugin>...</plugin>
    </plugins>
  </build>
</project>
```

Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 9 / 46 >

IntroductionMavenG8 et la forge G8LabIntégration continueMerge requestsRéférences

Projet Maven: dépendances dans le pom.xml

```
<project ...>
  <dependencies>
    <dependency>
      <groupId>org.openjfx</groupId>
      <artifactId>javafx-controls</artifactId>
      <version>11.0.2</version>
    </dependency>
  </dependencies>
  ...
</project>
```

En pratique, morceaux de XML à copier-coller depuis <https://search.maven.org/>, Maven s'occupe du reste !

Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 10 / 46 >

IntroductionMavenG8 et la forge G8LabIntégration continueMerge requestsRéférences

Repository Maven

- Dépôt contenant:
 - Des plugins
 - Des bibliothèques (en général Java)
- Sur le web
 - Téléchargement automatique à la demande
 - Défaut: <http://repo1.maven.org>
 - Miroirs (Nexus, Archiva, etc)
- Local: `~/.m2/repository`
 - contient les archives des projets locaux
 - ★ phase `install`
 - cache pour les *repository* web

Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 11 / 46 >

IntroductionMavenG8 et la forge G8LabIntégration continueMerge requestsRéférences

Classpath et dépendances

Utilisation de libs externes

- Téléchargement
- Gestion des versions (⇒ build reproductible)
- Transitivité des dépendances
- Configuration du CLASSPATH

Également utilisé pour les plugins

Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 12 / 46 >

IntroductionMavenG8 et la forge G8LabIntégration continueMerge requestsRéférences

Dépendances: scope

Dépendances dans le CLASSPATH dans chaque phase selon la portée (scope)

scope	compilation	test	exécution	déploiement
compile	x	x	x	x
provided	x	x	x	
runtime		x	x	x
test		x		

(+ system, import)
Exemple avec JUnit:

```
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-engine</artifactId>
  <version>5.7.0-M1</version>
  <scope>test</scope> <!-- not available in
                        'mvn exec:java' -->
</dependency>
```

Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 13 / 46 >

IntroductionMavenG8 et la forge G8LabIntégration continueMerge requestsRéférences

Archetypes

- Complexité inhérente aux projets maven
 - Difficultés de mise en œuvre
- Archetype = mini-projet de départ
 - D'un type particulier
 - Préconfiguré
- Exemples
 - `maven-archetype-quickstart`
 - `spring-mvc-jpa-archetype`
- En ligne de commande: `mvn archetype:generate`

Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 14 / 46 >

IntroductionMavenG8 et la forge G8LabIntégration continueMerge requestsRéférences

Intégration dans les EDI

Un bon IDE a besoin de connaître les dépendances (erreurs de typage, suggestions, ...).
`pom.xml` contient toutes les informations.

- IntelliJ : par défaut
- VSCode :
 - Plugin Maven for Java, inclus dans Java Extension Pack
 - File → Open Folder sur le répertoire contenant le `pom.xml` configure le projet automatiquement.
- Eclipse :
 - Plugin m2e + connecteurs
 - `mvn eclipse:eclipse`
 - ★ configure un projet Eclipse
 - ★ qui correspond au projet maven
- Netbeans: par défaut

Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 15 / 46 >

IntroductionMavenG8 et la forge G8LabIntégration continueMerge requestsRéférences

Forges logicielles

Outil de travail collaboratif pour le développement :

- Espace collaboratif
 - Partage de documents
 - Wiki
 - Dépôt (SVN, Mercurial, Git, etc)
- Gestion des tâches
 - Bug tracking
 - Support, tâches diverses
 - Calendrier, Gantt
 - Suivi via un système de tickets (Issues)

Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 16 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

forge.univ-lyon1.fr

- Forge Gitlab (logiciel libre installé sur nos serveurs)
- Dépôts Git (logiciel sur lequel repose GitLab, mais on peut utiliser Git sans GitLab)
- Intégration SI Lyon 1 (LDAP + CAS)
 - Disponible aux étudiants et personnels
 - Utilisable pour les TPs
 - Obligatoire pour le projet transversal (ex-MultiMIF)

Matthieu Moy (Transparents originaux d'Emmanuel Coqu)
Maven - Forge - Intégration continue
2020
< 17 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Git

- Gestionnaire de versions distribué
 - À la mercurial / darcs / bazaar / etc
- Utilisable
 - En ligne de commande (git)
 - Via une interface dédiée
 - ★ Tortoise git, SourceTree, etc
 - Dans un EDI
 - ★ Intégration VSCode, Eclipse, Netbeans, IntelliJ, Emacs, etc

Conseil : apprendre la ligne de commande (maîtriser les concepts, les noms des commandes), puis choisir l'outil qui vous convient.

Matthieu Moy (Transparents originaux d'Emmanuel Coqu)
Maven - Forge - Intégration continue
2020
< 18 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Commandes de base

- Création
 - init, clone
- Fichiers
 - add, remove, mv, status
- Versions
 - commit, checkout
- Branches
 - branch, merge, rebase
- Synchronisation de dépôts
 - pull, fetch, push

Matthieu Moy (Transparents originaux d'Emmanuel Coqu)
Maven - Forge - Intégration continue
2020
< 19 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Scénario simple

- 1 Début du travail
 - 1 Clone d'un dépôt distant
 - 2 Modification d'un fichier
 - 3 Commit
 - ★ Pour l'instant, pas de modification du dépôt distant
 - 4 Push vers le dépôt distant
- 2 Plus tard ...
 - 1 Pull du dépôt distant
 - ★ ou bien fetch + rebase (≈ git pull --rebase)
 - 2 GOTO 1.2

Matthieu Moy (Transparents originaux d'Emmanuel Coqu)
Maven - Forge - Intégration continue
2020
< 20 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Suivi des tâches (gestionnaire de tickets)

- Les tâches ont
 - Une description
 - Un statut: fermé ou ouvert
- Les tâches peuvent avoir:
 - Une échéance
 - Une personne assignée
 - Des étiquettes
 - Une étape (milestone)
- Liens commit (hash 32fb54de)/tâche (numéro 1234):
 - Fixes #1234 dans le message de commit ⇒ ferme le ticket + lien hypertexte
 - ref 32fb54de dans les commentaires de la tâche ⇒ lien vers le commit

Matthieu Moy (Transparents originaux d'Emmanuel Coqu)
Maven - Forge - Intégration continue
2020
< 21 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Branches: Why and How

- 1 branch = 1 named ref to a commit
- Think of a branch as a set of commits
- Typical uses
 - maintenance branch (bugfix only, will lead to next minor release) vs development branch (new features, will lead to next major release)
 - Topic branch: 1 branch per feature
 - ★ Create the branch
 - ★ Work on it (commit)
 - ★ Request a merge (push + pull-request, ...)
 - ★ (Delete the branch when it's merged)

Matthieu Moy (Transparents originaux d'Emmanuel Coqu)
Maven - Forge - Intégration continue
2020
< 22 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Branches and Tags in Practice

- Create a local branch and check it out:

```
git checkout -b branch-name1
```
- Switch to a branch:

```
git checkout branch-name
```
- List local branches:

```
git branch
```
- List all branches (including remote-tracking):

```
git branch -a
```
- Create a tag:

```
git tag tag-name
```

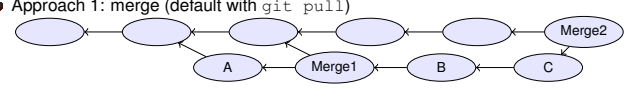
¹Since git 2.23, `git switch --create`

Matthieu Moy (Transparents originaux d'Emmanuel Coqu)
Maven - Forge - Intégration continue
2020
< 23 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Merging With Upstream

Question: upstream (where my code should eventually end up) has new code, how do I get it in my repo?

- Approach 1: merge (default with `git pull`)

- Drawbacks:
 - Merge1 is not relevant, distracts reviewers (unlike Merge2).

Matthieu Moy (Transparents originaux d'Emmanuel Coqu)
Maven - Forge - Intégration continue
2020
< 24 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
References

Merging With Upstream

Question: upstream (where my code should eventually end up) has new code, how do I get it in my repo?

- Approach 2: no merge

- Drawbacks:
 - In case of conflict, they have to be resolved by the developer merging into upstream (possibly after code review)
 - Not always applicable (e.g. "I need this new upstream feature to continue working")

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 25 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
References

Merging With Upstream

Question: upstream (where my code should eventually end up) has new code, how do I get it in my repo?

- Approach 3: rebase (git rebase or git pull --rebase)

- Drawbacks: rewriting history implies:
 - A', B, C, A', B' probably haven't been tested (never existed on disk)
 - What if someone branched from A, A', B or C?
 - Basic rule: don't rewrite published history

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 26 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
References

Rewriting history with rebase -i

- git rebase: take all your commits, and re-apply them onto upstream
- git rebase -i: show all your commits, and asks you what to do when applying them onto upstream:

```
pick ca6ed7a Start feature A
pick e345d54 Bugfix found when implementing A
pick c03fffc Continue feature A
pick 5bdb132 Oops, previous commit was totally buggy
```

```
# Rebase 9f58864..5bdb132 onto 9f58864
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
```

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 27 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
References

git rebase -i commands (1/2)

- p, pick** use commit (by default)
- r, reword** use commit, but edit the commit message
Fix a typo in a commit message
- e, edit** use commit, but stop for amending
 - Once stopped, use git add -p, git commit --amend, ...
- s, squash** use commit, but meld into previous commit
- f, fixup** like "squash", but discard this commit's log message
- Very useful when polishing a set of commits (before or after review): make a bunch of short fixup patches, and squash them into the real commits. No one will know you did this mistake :-).

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 28 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
References

git rebase -i commands (2/2)

- x, exec** run command (the rest of the line) using shell
 - Example: exec make check. Run tests for this commit, stop if test fail.
 - Use git rebase -i --exec 'make check'² to run make check for each rebased commit.

²Implemented by Ensimag students!

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 29 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
References

Git Reflog: TL; DR

Git's reflog = detailed history

(makes git rebase less dangerous)

Good news:
if you don't know how to use it, a Git expert around you may do and recover data you thought was lost :-)

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 30 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
References

Git's reference journal: the reflog

- Remember the history of local refs.
- ≠ ancestry relation.

- ref@{n}: where ref was before the n last ref update.
- ref~n: the n-th generation ancestor of ref
- ref^: first parent of ref
- git help revisions for more

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 31 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
References

Centralized workflow

```
do {
  while (nothing_interesting())
    work();
  while (uncommitted_changes()) {
    while (!happy) { // git diff --staged ?
      while (!enough) git add -p;
      while (too_much) git reset -p;
    }
    git commit; // no -a
    if (nothing_interesting())
      git stash;
  }
  while (!happy)
    git rebase -i;
} while (!done);
git push; // send code to central repository
```

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 32 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Triangular Workflow with pull-requests

- Developers pull from upstream, and push to a “to be merged” location
- Someone else reviews the code and merges it upstream

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 33 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Pull-requests in Practice

Contributor create a branch, commit, push

Contributor click “Create pull request” (GitHub, GitLab, BitBucket, ...), or `git request-pull`

Maintainer receives an email

Maintainer review, comment, ask changes

Maintainer merge the pull-request

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 34 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Code Review

- In a perfect world:
 - A writes code, commits, pushes
 - B does a review
 - B merges to upstream
- What usually happens:
 - A writes code, commits, pushes
 - B does a review
 - B requests some changes
 - ... then ?

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 35 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Iterating Code Reviews

- At least 2 ways to deal with changes between reviews:
 - Add more commits to the pull request and push them on top
 - Rewrite commits (`rebase -i, ...`) and overwrite the old pull request
 - The resulting history is clean
 - Much easier for reviewers joining the review effort at iteration 2
 - e.g. On Git's mailing-list, 10 iterations is not uncommon.

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 36 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Triangular Workflow: Advantages

- Beginners integration:
 - start committing on day 0
 - get reviewed later
- In general:
 - Do first
 - Ask permission after
- For Open-Source:
 - Anyone can contribute in good condition
 - “Who's the boss?” is a social convention

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 37 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Intégration Continue : l'idée

- Principe :
 - Intégrer le nouveau code au fur et à mesure qu'il est écrit
 - Garder une branche principale toujours fonctionnelle
- Outil indispensable : test automatisé à chaque `push`
- Abus de langage : « intégration continue » = « tests automatisés »

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 38 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Intégration Continue - contexte

Historiquement réservé aux projets

- de grande taille
- impliquant de nombreuses personnes
- avec des itérations courtes

Technologies actuelles → accessible sur de petits projets → une fois qu'on y a goûté, on en fait partout !

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 39 / 46 >

Introduction
Maven
Git et la forge GitLab
Intégration continue
Merge requests
Références

Intégration Continue - principes

- Automatisation des phases du cycle de vie
 - Compilation, test, mise à disposition de binaires
- Institutions de bonnes pratiques
 - Commit réguliers
 - La branche par défaut compile
 - ...
- Surveillance
 - Tableaux de bord, etc

Matthieu Moy (Transparents originaux d'Emmanuel Coq)
Maven - Forge - Intégration continue
2020
< 40 / 46 >


IntroductionMavenGit et la forge GitLabIntégration continueMerge requestsRéférences

Continuous Integration: example with GitLab-CI

<https://gitlab.com/moy/gitlab-ci-demo>

- Configuration (.gitlab-ci.yml):

```
junit:  
  # docker image with Maven  
  image: maven:3.6.1-jdk-11  
  script:  
    - cd my-project/ && mvn test --batch-mode
```
- Use: work as usual ;-)
 - Tests launched at each `git push`
 - Pass/failed indicator for each merge-request



Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 41 / 46 >

IntroductionMavenGit et la forge GitLabIntégration continueMerge requestsRéférences

Serveurs d'IC


Permet d'exécuter régulièrement:

- Checkout
- Compilation
- Test
- Audit de code

Pour gitlab:

- peut servir de serveur d'IC;
- nécessite de gérer et de configurer des "runners" qui exécuteront les tâches (merci Manu !)

Alternatives: Jenkins, Travis-CI, GitHub actions, etc




Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 42 / 46 >

IntroductionMavenGit et la forge GitLabIntégration continueMerge requestsRéférences

SonarQube

Outil d'audit de code

- Analyse exécutée lors du cycle de vie
 - Via e.g. un goal maven
- Fournit des tableaux de bord:
 - Qualité du code
 - Couverture des tests unitaires



Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 43 / 46 >

IntroductionMavenGit et la forge GitLabIntégration continueMerge requestsRéférences

Merge-requests (aka pull-requests sur GitHub)

- Principe:
 - Un contributeur fait un "fork" du projet
 - Il fait un `push` dans une branche perso
 - Il demande à intégrer (merge) son code
- Intérêts:
 - Liste des branches à intégrer facile à voir
 - Discussion sur le code avant intégration
 - Possibilité de rejeter le mauvais code



Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 44 / 46 >

IntroductionMavenGit et la forge GitLabIntégration continueMerge requestsRéférences

Tout ça ensemble?

- Début du travail:

```
git checkout -b mvc  
...  
git commit  
git push -u origin mvc
```
- Création de la pull-request sur l'interface web
- Gitlab-CI fait passer les tests :

```
# Fichier .gitlab-ci.yml  
...  
junit:  
  script:  
    - cd my-project/ && mvn test --batch-mode
```
- Maven s'occupe de télécharger les  tester, vérifier le style, et voilà :  updated 1 day ago



Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 45 / 46 >

IntroductionMavenGit et la forge GitLabIntégration continueMerge requestsRéférences

Références

- <https://maven.apache.org/>
- <https://git-scm.com/>
- <https://about.gitlab.com/>



Matthieu Moy (Transparents originaux d'Emmanuel CoquMaven - Forge - Intégration continue2020< 46 / 46 >

